A Multigraph Representation for Improved Unsupervised/Semi-supervised Learning of Human Actions

Simon Jones, Ling Shao
Department of Electronic and Electrical Engineering
The University of Sheffield, Sheffield, S1 3JD, UK
simon.m.jones@sheffield.ac.uk, ling.shao@sheffield.ac.uk

Abstract

Graph-based methods are a useful class of methods for improving the performance of unsupervised and semi-supervised machine learning tasks, such as clustering or information retrieval. However, the performance of existing graph-based methods is highly dependent on how well the affinity graph reflects the original data structure. We propose that multimedia such as images or videos consist of multiple separate components, and therefore more than one graph is required to fully capture the relationship between them. Accordingly, we present a new spectral method – the Feature Grouped Spectral Multigraph (FGSM) – which comprises the following steps. First, mutually independent subsets of the original feature space are generated through feature clustering. Secondly, a separate graph is generated from each feature subset. Finally, a spectral embedding is calculated on each graph, and the embeddings are scaled/aggregated into a single representation. Using this representation, a variety of experiments are performed on three learning tasks – clustering, retrieval and recognition – on human action datasets, demonstrating considerably better performance than the state-of-the-art.

1. Introduction

Graph-based algorithms are a powerful way of exploiting the underlying structure of a dataset to improve the performance of unsupervised and semi-supervised tasks. To illustrate this, we can consider three of the most successful graph-based methods: spectral clustering [8], which can be used to find unusually structured clusters; manifold ranking, which has been applied to information retrieval tasks with great success [3]; and Laplacian Eigenmaps (LE) [1], which are applied to dimensionality reduction. In general, by using graph-based methods, it is possible to uncover the latent structure of a high-dimensional dataset, thereby improving the accuracy of unsupervised and semi-supervised learning tasks.

The first step of these graph-based methods is to generate an affinity matrix, W, which represents the affinity between every pair of points in dataset X. For bag-of-features (BoF) histograms (which are the focus of this paper), it is possible to use a heat kernel applied to the χ^2 distance between every pair of points $x_i, x_j \in X$:

$$W_{ij} = exp\left(-\frac{\chi^2(x_i, x_j)}{\sigma^2}\right) \quad (1)$$

Alternatively, the histogram intersection could be used. After W is generated, various further operations are performed on W to get the final result. In certain methods, such as LE [1], W is made sparse using kNN or ϵ neighbourhoods, but in others it is fully connected.

Nonetheless, all graph-based methods for representation share the same flaw: when W is generated from X, there is significant information loss from the original feature space – only a single affinity value is generated for every pair of points. The information loss is particularly severe for small datasets (each row of W is therefore low-dimensional) or when the dataset's original feature space has a high dimensionality.

On high-dimensional data such as histograms, a single graph, generated using a single affinity metric, is not often sufficient to capture the full structure present in the original feature space. When representing realistic images or videos, there may be multiple statistically independent components within the histogram – then, a single graph would not be able to distinguish between these components. Instead, we suggest that multiple graphs should be constructed, each corresponding to a different component of the original images or videos.

In this paper, we present a novel method that generates multiple graphs from independent subsets of the

feature space. First, multiple graphs are found by partitioning the feature space into several mutually independent subspaces, then generating a different affinity matrix from each subspace. Then, a spectral embedding method is performed on each subspace. Finally, the embeddings are scaled and concatenated together, resulting in a single representation for each datapoint. This representation is referred to as the Feature Grouped Spectral Multigraph (FGSM). We expect FGSM to result in minimal information loss from the original feature space compared to ordinary spectral embedding methods. Through experimentation on several human action datasets, we demonstrate that FGSM gives superior results compared to the state-of-the-art algorithms for clustering, retrieval and recognition tasks.

The rest of this paper is structured as follows. Section 2 describes the theory and implementation details of the Feature Grouped Spectral Multigraph. Section 3 details our various experiments on clustering, retrieval, and recognition, and Section 4 concludes with a discussion of our findings.

2. Multigraph Representation

The analysis of human actions can come far in the past decade. While initial attempts to perform human action recognition have relied on global features such as silhouettes and HMM modeling, such methods have proven to be unreliable on the recent "realistic" human action datasets such as the ubiquitous Hollywood-2 [7]. Instead, most of the best performing works have focused on extracting local features, such as in Wang et al. [15] and Yang et al. [19], then constructing BoF histograms from these features.

When applying FGSM to a dataset, the original feature space of the dataset should have two properties: 1) the feature space must be high dimensional, and 2) the feature set must be divisible into several disjoint subsets with high independence between all the subsets and high dependence within the subsets. We propose that these properties apply to the histogram representation of videos due to the locality of the features – each histogram bin is primarily associated with a different component of the original video. This concept is illustrated in Figure 1.

In ordinary graph-based learning methods, much of the information from the original feature space will be lost in the creation of the affinity graph. FGSM, however, overcomes this issue by finding multiple independent views (subspaces) of the original representation and generating a separate affinity graph for each view.

The full algorithm for FGSM is shown in Algorithm 1.[1]

Algorithm 1: FGSM – Multigraph Representation

Data:
X - a histogram representation of a dataset
m - the number of feature subspaces to find
k - the number of eigenvectors per feature subspace
Result: Y - a multigraph representation of the dataset

1 Calculate HSIC affinity matrix between pairs of columns of X, where $W_{jk} = tr((L_j^T L_k)(L_k^T L_j))$ (Eqn 3)
2 Spectrally cluster W according to Ng et al. [8] to find m feature clusters: $C_1..C_m$
3 Define functions $P_1..P_m$ to project X into feature subspaces according to $C_1..C_m$
4 **for** $i \leftarrow 1$ *to* m **do**
5 Calculate $T \leftarrow P_i(X)$
6 Calculate $W_{jk} \leftarrow sum(min(T_j, T_k))$
7 Calculate $S \leftarrow D^{-1/2} L D^{-1/2}$, where $L \leftarrow D - W$ and D is a diagonal matrix with D_{ll} equal to the sum of the lth row of W
8 Find first k eigenvectors $e_1..e_k$ of S, concatenate them columnwise: $E_i \leftarrow [e_1..e_k]$
9 Normalise rows of E_i to sum to 1
10 Find λ_i as the mean distance between rows in M_i: $\lambda_i \leftarrow \sigma(dists(E_i))$
11 Concatenate scaled $E_1..E_m$ columnwise to get Y: $Y \leftarrow [(\lambda_1^{-1} E_1)..(\lambda_m^{-1} E_m)]$

2.1. Feature Grouping

The first step is to extract several mutually independent subspaces from the original feature space. This can be achieved by spectrally clustering features on an affinity graph of Hilbert-Schmidt Independence Criterion (HSIC) values, calculated between every pair of features.

HSIC captures all *non-linear* dependencies between two random variables x and y, as described in Gretton et al. [2], so long as the associated reproducing kernel Hilbert spaces are universal. It is more suitable for our purposes than other independence measures, such as the correlation co-efficient, which only capture linear dependencies. To demonstrate that it is a true independence measure, Gretton et al. show it equals zero if and only if x and y are independent. For our purposes,

[1] The MATLAB code for FGSM can be found at http://www.simonjonesdev.co.uk.

1 2 3 4 5 6

Figure 1: Certain BoF histogram bins may be associated more with one component of the video than any other. In this simplified 6-bin example, bins 1 and 2 are strongly associated with the upper body, bins 3 and 4 are strongly associated with the background, and bins 5 and 6 are strongly associated with the lower body. The high separability of the three components would make this histogram ideal for FGSM.

it can be empirically estimated from a finite number of (x_i, y_i) tuples by the following:

$$\rho_h(x, y) = \frac{1}{(1-n)^2} tr(HK_xHK_y) \quad (2)$$

where $H_{ij} = \delta_{ij} - n^{-1}$, K_x and K_y are the outer products of vectors x and y respectively, and n is the number of samples. Calculating K_x and K_y, however, takes $O(n^2)$ time and space, which is highly expensive for larger datasets, so incomplete Cholesky decomposition is used to find L_x and L_y, such that K_x and K_y can be approximated as $K'_x = L_x L_x^T$ and $K'_y = L_y L_y^T$. The approximate HSIC can then be calculated using the following:

$$\rho_h(x, y) = tr((L_x^T L_y)(L_y^T L_x)) \quad (3)$$

This completes in $O(nf^2)$ time, where f is the chosen number of columns in L. On very large datasets, HSIC estimation can be made more efficient by sparsely sampling the original population. Such sampling can be done with acceptable loss of accuracy, as the estimated HSIC approaches the true HSIC at speed $\frac{1}{\sqrt{n}}$.

To perform feature grouping, ρ_h is calculated on every pair of features i and j in the original feature space of dataset $\mathbf{x}$, resulting in affinity graph $W_{ij} = \rho_h(\mathbf{x}_i, \mathbf{x}_j)$. Spectral clustering is performed on W according to Ng et al. [8] to find m disjoint feature subspaces, $\mathbf{s}_1, .., \mathbf{s}_m \subset \mathbf{x}$. A large range of values for m give good results, as shown in experiments below, so this choice is not crucial. Nonetheless, $m \geq 20$ typically achieves the best results.

2.2. Multigraph Spectral Embedding

Having obtained m disjoint subspaces, it is possible to find m separate embeddings of the dataset according to each subspace. For each subspace s_m, an affinity graph W is constructed using:

$$W_{ij} = sum(min(P_m(x_i), P_m(x_j))) \quad (4)$$

where $P_m(x)$ is a function that maps x to the mth subspace. Rather than using a kNN-neighbourhood or a ϵ-neighbourhood graph, as typically used in Laplacian Eigenmaps, W is constructed as a fully connected graph as in Ng et al.[8]. The choice to use a fully connected graph is made empirically – in preliminary experiments, a fully connected graph gave better results than a kNN neighbourhood graph for any k.

Spectral embedding is then performed on W as per steps 2-4 of Ng et al. to find a spectral embedding. These steps are:

1. Find $L = D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix with D_{ii} equal to the sum of the ith row of W.
2. Find the k highest eigenvectors of L, $e_1, .., e_k$, and construct a matrix E columnwise as $[e_1..e_k]$.
3. Normalise E so each row sums to 1.

It is notable that this process also differs from Laplacian Eigenmaps, because of step 3, instead follows Ng et al.[8]. The unit normalisation is important to reduce the scale variation between the m separate embeddings. The optimal choice of k is likely to vary between spectral embeddings – however, for simplicity a single k is chosen that is uniform across all embeddings. Future work might show improved performance heuristically choosing an individual k per embedding.

The final step to generate the FGSM is to aggregate the m embeddings. This can be simply and naively achieved by concatenating all $E_1, .., E_m$ columnwise: $X = [E_1..E_m]$. Then, row i of X is an $m \times k$ length vector describing sample x_i. While this scheme works well, however, further performance increases can be achieved by scaling each embedding appropriately before aggregation. The Euclidean distance is calculated between every pair of rows in E_i and it is used to find λ_i thus:

$$dists_{i,jk} = ||e_{i,j} - e_{i,k}||^2 \quad (5)$$

$$\lambda_i = \sigma(dists_i) \quad (6)$$

where $\sigma(x)$ is the standard deviation of the values in x. Then, to get our final representation, scale each $E_1, .., E_m$ with $\lambda_1, ..., \lambda_m$ and concatenate columnwise: $X = [(\lambda_1^{-1}E_1)..(\lambda_m^{-1}E_m)]$. As a result, each embedding is scaled to have a total distance variation of 1.

We do not consider out-of-sample extension in this paper, which could be necessary when performing time critical tasks such as recognition or retrieval. However, the Nyström approximation could be easily applied to each embedding separately, as in [14], to achieve out-of-sample extension.

3. Experiments

In this section FGSM is applied to various machine learning problems to demonstrate its applicability to real-world machine learning tasks. We specifically consider several realistic human action datasets, although in future work FGSM could also be applied to image datasets.

3.1. Datasets

Four real-world datasets are used for experimentation, and are described below.

The UCF YouTube [5] dataset has 1168 videos of human activities that have been gathered from YouTube, and are of low resolution/quality. The actions are in 11 classes.

The UCF Sports [11] dataset has 150 sports videos recorded from broadcast. There are a total of 13 action categories recorded from consistent angles.

The Hollywood-2 [7] dataset has 1707 action videos gathered from Hollywood movies. There are 12 action categories. The actions vary greatly in appearance, with many instances of partial occlusion, unusual viewpoints, different actors, and varying execution of the actions.

The Olympic Sports [10] dataset consists of 783 action videos collected from TV footage of olympic sports. There are 16 action classes. This dataset has more view variation than the UCF Sports dataset, and is correspondingly more difficult.

3.2. Setup

To extract a motion representation of the actions, the publicly available code for dense trajectory feature extraction is used as presented in Wang et al. [15], with the default settings of the software. This results in a series of local features, each feature represented with 4 descriptors (HOG, HOF, MBH, Tr). PCA, then k-means, are performed on each of the 4 descriptors in turn. For k-means, $k = 4000$. A separate histogram is generated for each descriptor, and then the histograms are aggregated together. The result is a 16000-bin histogram per video. When performing FGSM, we grouped features in 30 subspaces (i.e., $m = 30$) and extracted 40 eigenvectors for each manifold ($k = 40$).

We compare our method for clustering features to the diffusion map method presented in Liu et al. [6], which is used for finding semantic words in bag-of-words models. To create a method for comparison, steps 1-3 of Algorithm 11 are replaced with the representation and clustering algorithm given in section 4 of [6]. The parameters for diffusion maps are optimised empirically. In our experimental results, this hybrid algorithm is referred to as DM (for Diffusion Maps), and we compare it to FGSM both for clustering and retrieval below.

3.3. Clustering

We first consider action clustering. Given an action dataset with k action classes, the goal is to find k disjoint subsets of the action dataset so that each subset contains only one action class. Wang et al. [17] introduced the concept of human action clustering, applying spectral clustering to features extracted from images of actions. Niebles et al. [9] use methods from document analysis to cluster actions based on their latent topics. Yang et al. [19] create a highly invariant feature extractor which can be used for effective clustering and one-shot learning. However, all of these methods either focus on improving the feature extraction process, or apply existing clustering methods from another domain to human actions. The FGSM representation can improve clustering on human actions by finding a strongly representative low-dimensional embedding of the original histograms.

To measure a clustering algorithm's performance in this paper, we use the same performance metric as in [19]. If each cluster c contains datapoints $x_1, .., x_n$, and each datapoint is associated with a ground truth label $l_1, .., l_n$, the label l_c of cluster c is determined to be:

$$\arg\max_{l_c} \sum_{i=1}^{n} \begin{cases} 1 & \text{if } l_c = l_i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The accuracy is then percentage of data points across the whole dataset that have the same label as their assigned cluster.

The results of various clustering methods are in Table 1. SC1 and SC2 are the methods presented in Shi and Malik [13] and Ng et al. [8], respectively, and they differ in how they calculate the normalised graph Laplacian. For both SC1 and SC2, the affinity matrix W is generated using the histogram intersection in Equation 4. For FGSM, the FGSM representation is

Dataset	Clustering Accuracy (%)			
	SC1	SC2	DM	FGSM
YouTube	22.3	39.2	39.9	**42.6**
UCF Sports	32.6	68.0	27.6	**70.8**
Hollywood-2	18.2	33.6	34.5	**38.6**
Ol. Sports	23.1	39.7	40.3	**42.8**

Table 1: Clustering performance of various methods on each dataset.

applied to the dataset, followed by ordinary k-means clustering using the Euclidean distance. Each clustering algorithm is run for 100 trials and the mean accuracy over all results is shown.

As can be seen from the table, FGSM achieves superior results to any of the compared methods on all four datasets. As stated above, this is likely because the single graph in SC1/SC2 is unable to capture all of the information in the original feature space, and because DM's feature clustering is not as accurate as ours. The UCF Sports dataset results are improved the least by FGSM (2.8% over ordinary spectral clustering) whereas in the Hollywood-2 dataset a 5% accuracy boost is observed. The scale of the improvement appears to be related to the size of the dataset – the larger the dataset, the larger the improvement that FGSM gives. This intuitively makes sense, as spectral embedding methods tend to be more accurate for larger populations. Performing the initial feature grouping may also be more accurate on larger datasets.

3.4. Content-based Retrieval with Relevance Feedback

Next, we consider content-based video retrieval (CBVR). This is recently a popular research field, although the bulk of retrieval work is on images rather than videos. Typically CBVR is aimed at improving the accuracy of multimedia search engines.

The formal aim of CBVR is as follows: given a query video, rank the videos in a video database according to their relevance to the query, and return the most relevant videos. Once a query has been submitted and the results returned, a user can give relevance feedback to the system, by marking each result item as "positive" or "negative", indicating whic h results are related to the query or not. The CBVR system can incorporate this relevance feedback to perform a further query, and return improved results.

Previous works, such as [4] and [12] have performed human action retrieval, representing actions using local features and performing relevance feedback to improve

Data	RF	Accuracy of Top 20 Results (%)				
		HI	CD	MR	DM	FGSM
YT	B	53.2	52.4	56.4	56.6	**63.5**
	A	74.4	74.4	75.2	75.9	**82.8**
U Sp.	B	38.5	38.3	39.5	17.9	**41.5**
	A	48.0	48.6	49.0	28.5	**51.7**
HW-2	B	25.4	25.2	28.1	26.2	**30.4**
	A	41.0	41.1	42.7	33.9	**46.0**
Ol. Sp.	B	35.2	34.7	37.7	39.4	**41.5**
	A	51.9	51.4	51.4	54.4	**59.6**

Table 2: Retrieval performance of various methods on each dataset, before (B) and after (A) relevance feedback (RF).

results. However, the most effective retrieval method to date was applied to image retrieval: manifold ranking, presented in He et al. [3], which is a graph-based algorithm. Manifold ranking incorporates the underlying structure of the dataset to rank the database items according to their similarity to the query. It can also elegantly incorporate positive and negative feedback to improve its rankings further. Recent work [18] has shown how manifold ranking can be made efficient enough for practical use on retrieval tasks. However, the rankings are generated from a single graph, so FGSM is likely to outperform it.

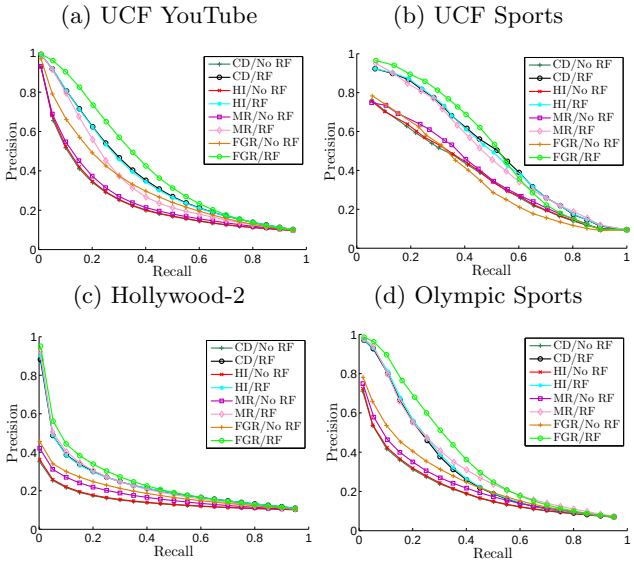

Figure 2: Precision/recall curves for video retrieval on various datasets.

We compare the performance between histogram intersection ranking (HI), χ^2 distance ranking (CD),

manifold ranking (MR) Liu et al's method [6] (DM)and FGSM ranking (FGSM) in Table 2, showing retrieval performance before and after a single round of relevance feedback. Ranking based on one-manifold embedding was considered but is not included in the results above for reasons of space – in preliminary experiments it uniformly performed worse than manifold ranking. To test, we set each video in the dataset as the query in turn, performing retrieval on the remaining videos. We determine the percentage of relevant videos in the top 20 results, and average this over all the queries.

To simulate relevance feedback, we mark several of the top 20 results as positive or negative according to the ground truth of the dataset, and rerun the query. For HI, CD, and FGSM we incorporate relevance feedback using a kernel SVM. For HI, we use the histogram intersection kernel; for CD, the χ^2 distance; for FGSM, the RBF kernel. To incorporate positive/negative relevance feedback in manifold ranking, we use scheme 1 as presented in [3], setting $\gamma = 0.25$.

As shown in the table, FGSM performs well for retrieval, especially after relevance feedback. A multigraph representation confers an advantage over manifold ranking (MR) method and DM, and gives the best performance for all four datasets. In Figure 2 we also show the precision-recall curves for each dataset, showing the clear advantage of FGSM.

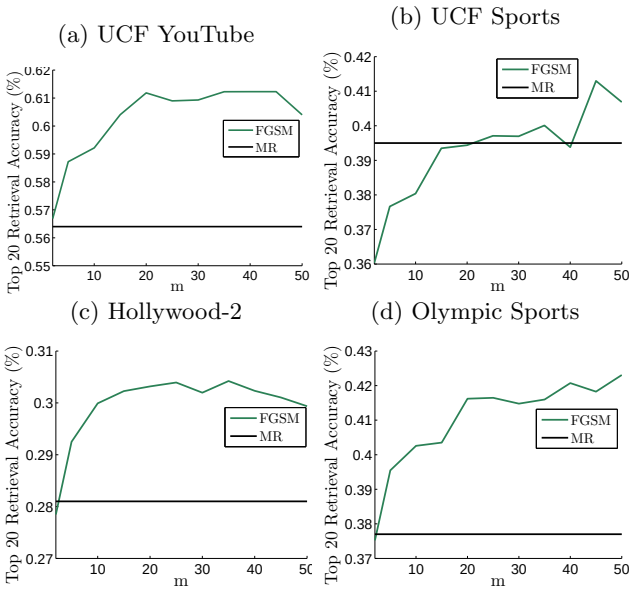

Figure 3: The effect of varying parameter m on retrieval performance in the top 20 results, versus manifold ranking baseline.

Finally, in Figure 3 we show the effects of varying parameter m on retrieval performance, compared against baseline manifold ranking performance. As can be seen from the figure, performance is vastly increased even when $m = 2$, and continues to rise until about $m = 30$ for all datasets. Performance is weakest compared to manifold ranking on the UCF Sports dataset, perhaps due to the small number of videos in that dataset.

Dataset	Recognition Accuracy (%)			
	Orig	STP	LE	FGSM
YouTube	84.1	85.4	74.6	**89.0**
UCF Sports	88.0	**89.1**	67.6	87.7
Hollywood-2	58.2	**59.9**	45.8	58.2
Ol. Sports	74.1	**77.2**	57.5	74.6

Table 3: Recognition performance of various methods on each dataset.

3.5. Recognition

Our final experiment is fully-supervised action recognition using FGSM. Here, we do not expect to outperform the state-of-the-art. Spectral embedding methods such as FGSM perform well on unsupervised and semi-supervised tasks because they make use of latent structural information in the unlabeled portion of the dataset. When performing fully supervised action recognition, however, all of the training data are labeled – it is not necessary to find the latent structure of fully labeled data. Instead, a discriminative classifier such as a kernel SVM can use all of the data to accurately model the separating hyperplane between classes even on the original feature space. A spectral embedding method will lose much information from the original feature space, making an optimal hyperplane between classes *harder* to find.

Instead of outperforming the state-of-the-art, we only intend to show that our method does not result in significant *loss* of recognition accuracy compared to the original representation, thus demonstrating that FGSM retains all important components from the original feature space.

In Table 3, FGSM is compared against the state-of-the-art human action recognition work in Wang et al. [16]. Orig is the dense trajectory histogram method presented in [16], using a multi-channel chi^2 kernel SVM for classification; STP is the spatio-temporal pyramid representation in [16] using a multi-channel chi^2 kernel SVM for classification; LE applies Laplacian Eigenmaps to the histogram and uses an RBF kernel SVM for classification; FGSM is the multigraph representation presented in this paper. To evaluate

classification with FGSM, we apply a kernel SVM to the FGSM representation of a dataset - we determined empirically that an RBF kernel works better than a linear or quadratic kernel. For each dataset, we use the same experimental setup provided in Wang et al. [16]

As can be seen, on recognition tasks FGSM consistently outperforms Laplacian Eigenmaps, and performs similarly to the original histogram representation Orig. This illustrates that our FGSM preserves the underlying structure of the dataset far better than a single spectral embedding. For one of the datasets – YouTube – FGSM even surpasses the state-of-the-art results in Wang et al., which is a surprising result, requiring further investigation. STP achieves the best results on the other three datasets, as it takes into account the spatio-temporal structure of the videos – in future, it may be possible to achieve even better results by combining STP with FGSM.

4. Discussion

In this paper we have introduced a new method for representing multimedia data – particularly human actions – for improved accuracy in clustering, retrieval and recognition tasks. Based on previous works on spectral embedding, we generated several spectral embeddings on separate subspaces of the original feature space, postulating that this would maximise the retained information from the original feature space. Through comprehensive experiments on four datasets, we have demonstrated that our new representation – FGSM – can surpass the state-of-the-art for clustering and retrieval/relevance feedback tasks on all datasets, and can also surpass the state-of-the-art recognition accuracy on certain datasets.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2002. 1

[2] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Proc. Algorithmic Learning Theory*, pages 63–77. Springer-Verlag, 2005. 2

[3] J. He, M. Li, H. jiang Zhang, H. Tong, and C. Zhang. Manifold-ranking based image retrieval. In *In ACM Multimedia*, pages 9–16, 2004. 1, 5, 6

[4] S. Jones and L. Shao. Content-based retrieval of human actions from realistic video databases. *Information Sciences*, 236:56 – 65, 2013. 5

[5] J. Liu, J. Luo, and M. Shah. Recognizing Realistic Actions from Videos "in the Wild". In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 1996–2003, June 2009. 4

[6] J. Liu, Y. Yang, I. Saleemi, and M. Shah. Learning semantic features for action recognition via diffusion maps. *Comput. Vision and Image Understanding*, 116(3):361–377, 2012. 4, 6

[7] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 2929–2936. IEEE, June 2009. 2, 4

[8] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances Neural Inform. Process. Syst.*, pages 849–856, 2001. 1, 2, 3, 4

[9] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *Int. J. Comput. Vision*, 79(3):299–318, Mar. 2008. 4

[10] J. C. Niebles, C. wei Chen, and L. Fei-fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc. European Conf. Comput. Vision*, pages 392–405, 2010. 4

[11] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2008. 4

[12] L. Shao, S. Jones, and X. Li. Efficient search and localization of human actions in video databases. *IEEE Trans. Circ. Syst. Video Tech*, 24(3):504–512, Mar 2014. 5

[13] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997. 4

[14] A. Talwalkar, S. Kumar, and H. A. Rowley. Large-scale manifold learning. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2008. 4

[15] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 3169–3176, 2011. 2, 4

[16] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vision*, 103:60–79, 2013. 6, 7

[17] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 1654–1661, 2006. 4

[18] B. Xu, J. Bu, C. Chen, D. Cai, X. He, W. Liu, and J. Luo. Efficient manifold ranking for image retrieval. In *ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 525–534, 2011. 5

[19] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1635–1648, 2013. 2, 4